Introducere în JavaScript

Introducere

- JavaScript *nu face parte* din familia Java
  - Numele original al JavaScript a fost “LiveScript”
  - Numele a fost schimbat când Java a devenit popular
  - Unele aspecte sintactice seamana pentru ca ambele limbaje au adoptat multe caracteristici ale limbajului C
- JavaScript este destinat programelor de mici dimensiuni

Utilizare în browser

- Codul JavaScript este inclus printr-un element <script>:

```
<script type="text/javascript">
    <!--
        document.write("Hello World!");
    //-->
</script>
```

- Obs.:
  - atributul type permite utilizarea si a altor limbaje de script (JavaScript este cel implicit)
  - Utilizarea “;” la sfârsitul unei instructiuni este optionala (necesara doar pentru a desparti mai multe instructiuni scrise pe o aceeasi linie)
  - Delimitatorii <!-- … //--> transmit navigatoarelor care nu suporta Javascript sa ignore continutul elementului <script>

Locul Javascript

- În sectiunea **<head>** a unui document HTML vor fi plasate functiile JavaScript (astfel, acestea vor fi încarcate înainte de a fi necesar)
- În sectiunea **<body>** vor fi inclusele fragmentele Javascript care se doresc executate la încarcarea paginii
- Poate fi utilizat un fisier .js separat (care **nu** va mai include tagul **<script>**). În documentul HTML:
  <script src="myJavaScriptFile.js"></script>
- Instructiuni JavaScript pot fi incluse si în cadrul unui element de formular HTML (de ex., buton)
  – Vor fi executate în mometul utilizarii elementului respectiv

Tipuri primitive de date

- 3 tipuri “primitive” de date: **number**, **string** si **boolean**
  - Orice altceva este considerat object
- Numerele sunt tratate ca având valori reale
- Sirurile pot fi delimitate de ghilimele “ sau de apostrofuri ‘
  - Pot contine caractere speciale **\n**, **\"** etc.
- Valorile booleene sunt **true** sau **false**
  - 0, "0", sirul vid, undefined, null si NaN sunt evaluate la false , orice alta valoare la true

Variabile

- Variabilele sunt declarate cu mentiunea optionala (dar recomandata) **var**:
  - **var** pi = 3.1416, x, y, nume = “Web" ;
  - Numele lor încep cu litera sau _ si sunt case-sensitive
  - Variabilele nu au tip: primesc tipul valorii care le este asignata la momentul curent
- Comentariile – ca în C sau Java:
  - Între ***//*** si sfârsitul liniei curente
  - Între ***/**** si *****/***

Operatori

- Aritmetici: **+ - * / % ++ --**
- De comparare:
  < <= == != >= >
- Logici: **&& || !**
- Pe biti: **& | ^ ~ << >>**
- De asignare: **+= -= *= /= %=**
  <<= >>= >>= &= ^= |=

Operatori

- Concatenare de siruri: **+**
- Operator conditional:
  condition ? value_if_true : value_if_false
- Testarea egalitatii:
  == si **!=** convertesc operanzii la acelasi tip înainte de testarea egalitatii
  === si **!==** considera operanzii inegali daca sunt de tipuri diferite
- Operatori aditionali:
  new **typeof** **void** **delete**

typeof

- Operatorul unar **typeof** returneaza unul din urmatoarele rezultate: "number", "string", "boolean", "object", "undefined" sau "function"
  - typeof null este "object"
  - Daca stud este un tablou, typeof stud este "object"

Instructiuni

- Atribuire: **salut = "Buna," + nume;**
- Instructiune compusa:

  { instr; ...; instr }
- If statements:

  if (cond) instr;

  if (cond) instr; else instr;
- Familiar loop statements:

  while (cond) instr;

  do instr while (cond);

  for (initializare; conditie; incrementare)

  instructiune;

Instructiuni

- **switch (expresie) {**
  case eticheta:
  instr;
  break;
  case eticheta:
  instr;
  break;
  ...
  default: instr;
  }
- Alte instructiuni:
  - **break;**
  - **continue;**
  - Instructiunea vida: **;;** sau **{ }**

Tablouri | 4 metode de creare

- Declaratie de tablou:
  var colors = ["red", "green", "blue"];
- Instantiere **new Array()** pentru a crea un sir vid:
  - **var colors = new Array();**
  - Valorile vor putea fi asignate mai târziu:
    colors[0] = "red"; colors[2] = "blue";
    colors[1]="green";
- Precizarea dimensiunii tabloului: new Array(n)
  - **var colors = new Array(3);**
- Utilizarea new Array(...) cu unul sau mai multe argumente indicând valorile elementelor:
  - **var colors = new Array("red","green", "blue");**

Dimensiunea unui tablou

- Daca numele unui tablou este **colors** lungimea îi este data de **colors.length**
- Dimensiunea unui tablou poate fi modificata peste limita curenta, prin instr. de atribuire:
  - Ex: var vizitatori = new Array(5); vizitatori[10] = 3;
- Tablourile au proprietatea sparse, este alocat spatiu doar pentru elementele carora le-a fost asignata o valoare
  - Ex: vizitatori[50000] = 3; - OK
  - Indicii trebuie sa fie între 0 si 2^{32}-1
- Nu exista tablouri bidimensionale, dar pot fi declarate tablouri de tablouri: vizitatori[5][3]

Functii pt. tablouri

Daca studenti este un tablou,

- myArray.sort() sorteaza tabloul alfabetic
- myArray.reverse() inverseaza elements
- myArray.push(...) adauga un numar de oricâte elemente noi la sfârsitul tabloului, incrementându-i dimensiunea
- myArray.pop() elimina si returneaza ultimul element dintr-un tablou, decrementându-i lungimea
- myArray.toString() returneaza un sir continând valorile elementelor tabloului, separate prin virgula

Instr. for ... in

- Permite traversarea tuturor proprietatilor unui obiect:
- **for (variabila in obiect) instr;**
  - Ex:
    ```
    for (var prop in course) {
        document.write(prop + ": " + course[prop]);
    }
    ```
  - Rezultat posibil:
    ```
    teacher: MBrut
    title: We Design
    ```
  - Proprietatile sunt accesate într-o ordine nedeterminata
  - Tablourile sunt obiecte ⇒ aplicata unui tablou, for...in va traversa “proprietatile” 0, 1, 2, ...
  - Obs.: course["teacher"] ⇔ course.teacher

Functii pt. siruri

- charAt(n)
  - returneaza al n-lea caracter din sir
- concat(string1, ..., stringN)
  - concateneaza sirurile date ca parametru, returnând rezultatul în sirul prin care este apelata metoda
- indexOf(substring)
  - returneaza pozitia primului caracter al lui ***substring*** în sirul prin care e apelata functia, sau -1
- indexOf(substring, start)
  - returneaza pozitia primului caracter al lui ***substring*** care începe la sau dupa pozitia ***start*** în sirul de apel, sau -1
- lastIndexOf(substring), lastIndexOf(substring, start)
  - similar cu indexOf, dar cautarea începe de la sfârsitul sirului

Referinte HTML

- Într-un doc. HTML, **window** este obiectul global
  - Toate variabilele sunt proprietati ale acestui obiect, sau ale unor obiecte descendente
  - Cea mai importanta proprietate este **document**
- Elementele **form** pot fi referite prin **document.forms[nrform].elements[nrElem]**
- Atributul name al elementului form poate fi utilizat ca referinta de tablou
  - Daca <form name="myForm">
    <input type="button" name="myButton" ...>
  - Atunci, în loc de document.forms[0].elements[0]
  - poate fi folosit document.myForm.myButton

Javascript si DOM

- JavaScript utilizeaza un Document Object Model (DOM) care descrie structura unei pagini Web
  - Nu este acelasi model ca XML DOM
- Utilizare:
  - Accesarea elementelor dintr-o pagina Web
  - Capturarea evenimentelor
  - Efectuarea dinamica de modificari într-o pagina

Javascript si DOM

- Unele elemente (e.g. <form>) HTML pot raspunde la actiuni utilizator prin crearea de evenimente

```
<form method="post" action="">
 <input type="button"
        name="myButton"
        value="Click me"
        onclick="alert('You clicked the button!');">
</form>
```

Evenimente

- Majoritatea elementelor HTML produc urmatoarele eveninemte:
  - onClick
  - onDblClick
  - onMouseDown – butonul mouse-ului este apasat în timp ce este plasat deasupra elementului
  - onMouseOver
  - onMouseOut
  - onMouseUp
  - onMouseMove
- În JavaScript, numele acestor evenimente vor fi scrise cu litere mici

Exemple

- <h1 onMouseOver="style.color='red';" onMouseOut="style.color='blue';">Hello </h1>
- “Rollover” de imagini:

```
<img src="../Images/coperta1.gif"
     width="55" height="68"
     onMouseOver="src='../Images/coperta2.gif';"
     onMouseOut="src='../Images/coperta1.gif';">
```

Evenimente

- http://developer.netscape.com/docs/manuals/js/client/jsguide/index.htm

Event	Applies to	Occurs when	Handler
Load	Document body	User loads the page in a browser	onLoad
Unload	Document body	User exits the page	onUnload
Error	Images, window	Error on loading an image or a window	onError
Abort	Images	User aborts the loading of an image	onAbort

Evenimente

Event	Applies to	Occurs when	Handler
KeyDown	Documents, images, links, text areas	User depresses a key	onKeyDown
KeyUp	Documents, images, links, text areas	User releases a key	onKeyUp
KeyPress	Documents, images, links, text areas	User presses or holds down a key	onKeyPress
Change	Text fields, text areas, select lists	User changes the value of an element	onChange

Evenimente

Event	Applies to	Occurs when	Handler
MouseDown	Documents, buttons, links	User depresses a mouse button	onMouseDown
MouseUp	Documents, buttons, links	User releases a mouse button	onMouseUp
Click	Buttons, radio buttons, checkboxes, submit buttons, reset buttons, links	User clicks a form element or link	onClick

Evenimente

Event	Applies to	Occurs when	Handler
MouseOver	Links	User moves cursor over a link	onMouseOver
MouseOut	Areas, links	User moves cursor out of an image map or link	onMouseOut
Select	Text fields, text areas	User selects form element's input field	onSelect

Evenimente

Event	Applies to	Occurs when	Handler
Move	Windows	User or script moves a window	onMove
Resize	Windows	User or script resizes a window	onResize
DragDrop	Windows	User drops an object onto the browser window	onDragDrop

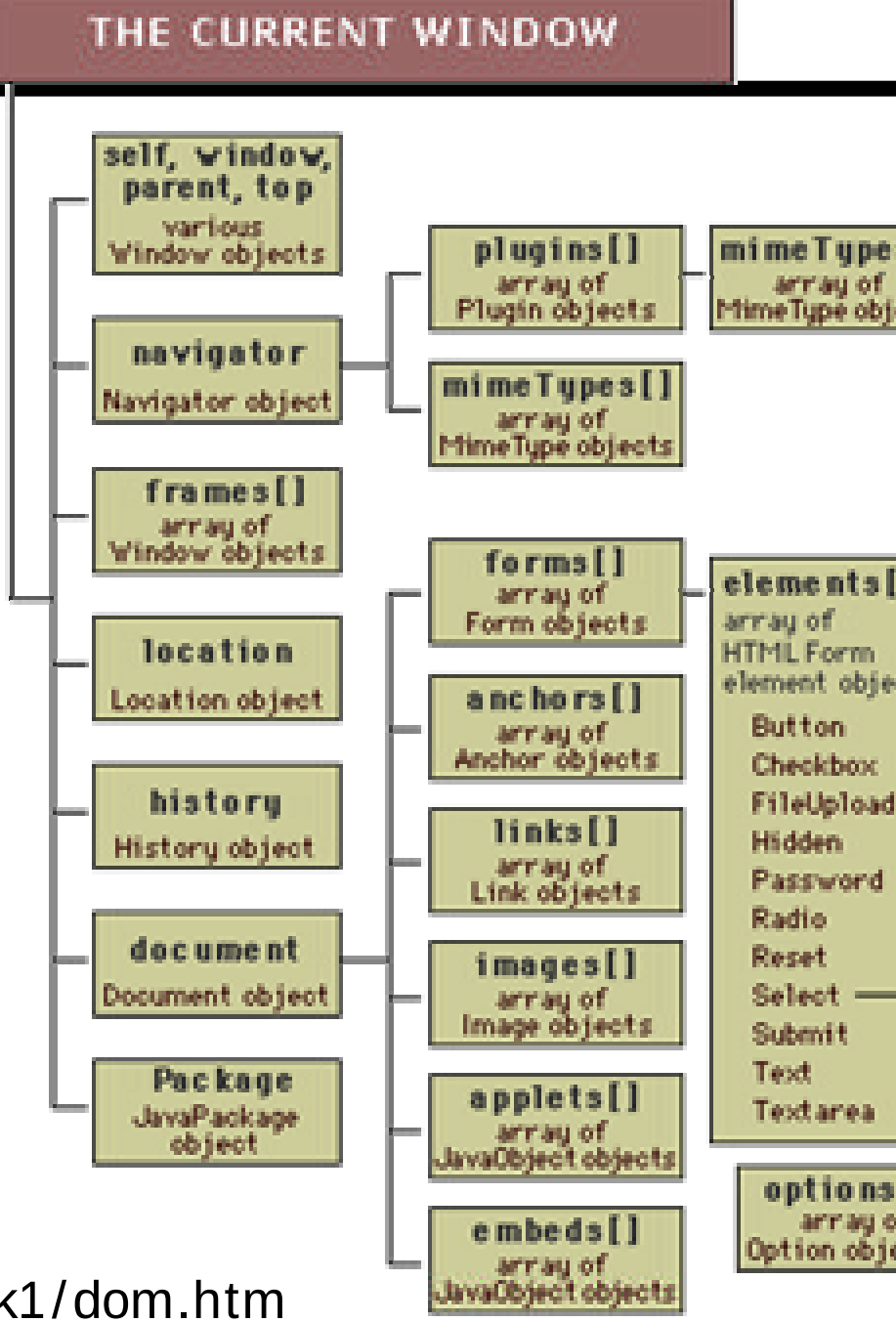
Evenimente

Event	Applies to	Occurs when	Handler
Focus	Windows and all form elements	User gives element input focus	onFocus
Blur	Windows and all form elements	User moves focus to some other element	onBlur
Reset	Forms	User clicks a Reset button	onReset
Submit	Forms	User clicks a Submit button	onSubmit

Javascript si DOM

- DOM-ul de baza este un standard W3C
- Elementul de cel mai înalt nivel pentru pagina curenta este window, orice alt element este descendent al acestuia
- Orice variabila Javascript este proprietate a unui obiect

Ierarhia DOM



Sursa:
http://sislands.com/coin70/week1/dom.htm

window | Proprietati

- window – HUHDVWD FXUHQW (implicit).
- self – sinonim cu window.
- parent – în cadrul unui frame, fereastra imediat urmatoare.
- top – cea mai exterioara fereastra.
- frames[ ] un tablou al frame-urilor din fereastra curenta. Frame-urile sunt ele însele ferestre.
- length numarul de frame-uri continute

window | Proprietati

- document
  - documentul HTML care va fi afisat în fereastra.
- location
  - adresa URL a documentului HTML care va fi afisat în fereastra. Apelul location.reload() va determina reîncarcarea

window | Proprietati

- open(URL)
  - Deschide o noua fereastra, continând documentul specificat prin URL.
- close()
  - Închide fereastra specificata (care nu trebuie sa fie un frame).

document | Proprietati

- Aceste proprietati vor trebui precedate de document.
- anchors[ ]
  - Un tablou al obiectelor Anchor din documentul curent (reprezentând elementele <a name=...>)
- forms[ ]
  - Un tablou al obiectelor Form
    - daca exista un singur formular, va fi referit prin forms[0]

document | Proprietati

- images[ ]
  - Un tablou al obiectelor Image
    - Pt. a VFLP EDLP DJ LQHD , se asigneaza un nou URL proprietatii src
- links[ ]
  - Un tablou al obiectelor Link
    - href este printre proprietati

document | Proprietati

- bgColor
  - Culoarea de fundal a documentului
    - Poate fi modificata în orice moment
- title
  - Un sir read-only continând titlul documentului
- URL
  - Un sir read-only continând URL-ul documentului

form | Proprietati

- elements[ ]
  - Un tablou al elementelor formularului

Posibilitati de debug

- Mozilla

  Tools Þ Options Þ Web Features Þ Enable JavaScript
- Netscape 6:

  Tasks Þ Tools Þ JavaScript console
- Netscape 4:

  Communicator Þ Tools Þ JavaScript console
- Orice versiune Mozilla sau Netscape:
  - **javascript:** în bara de adresa, apoi Enter
- Despre facilitare Javascript în diferite navigatoare:

  http://www.valleyvet.com/si_javascript_help.html

Întrebari?